

Ingeniería del Software II

Curso 2017-2018

Enzo Ferey - enzo.ferey@alumnos.upm.es

Tema 1 - Introducción al trabajo en equipo

Un equipo es un grupo de al menos 2 personas que trabajan hacia un objetivo común. Cada persona tiene un rol específico asignado y la terminación de la misión requiere alguna forma de dependencia entre los miembros del equipo.

Los fallos más comunes en un equipo son:

- Liderazgo ineficaz
- Fallo al comprometerse o cooperar
- Falta de participación
- Falta de confianza
- Pobre calidad
- Añadir funciones a los requisitos para mejorar sus productos

Roles en reunión

Líder: es la persona que dirige el equipo. Convoca y facilita reuniones. Orquesta todas las actividades del equipo. Interesado en la resolución de los problemas.

Facilitador: mantiene la discusión enfocada en el punto y en movimiento. Impide con mucho tacto que alguien sea dominante o pase desapercibido y ayuda a concluir las discusiones.

Cronometrador: notifica y avisa al grupo cuando el tiempo asignado para un punto ha pasado o está a punto de terminar.

Notario: persona que toma nota de los temas principales, puntos claves que se trataron y decisiones tomadas. Elabora el acta de la reunión.

Evaluable final: se evalúa el rendimiento de la reunión para mejorar las próximas.

Es fundamental disponer de una agenda en toda reunión. He abajo una posible plantilla para éstas:

Convocada por: G. Cuevas

Fecha: 1/12/03

Hora de inicio: 9:00 a.m

Lugar: Sala Auditorio

Hora de acabar: 11:00 a.m.

Asunto: Reunión de Planificación: Encuesta de Empresas

Hora	Puntos del días	Participantes	Resultado esperado
9:00 a.m.	Revisión de la agenda	Todos	Modificar agenda si es necesario
9:10	Propósito y resultado esperado de la encuesta de empresas	G.Cuevas*, M. García	Desarrollar una definición del propósito del proyecto y del resultado esperado
9:30	Alcance del trabajo, calendario y recursos requeridos	G. Cuevas*, M. García, A. Amescua	Hacer un Borrador con el alcance general del trabajo, calendario y los recursos necesarios
9:50	Planificación detallada: licitación, empresas candidatas, análisis de información, recursos humanos, etc.	Miguel García*, A. Amescua, J.A. Cerrada, J. García	Desarrollar una descripción preliminar de los por menores del proyecto
10:10	Siguientes pasos	G. Cuevas*, todos	Desarrollar acciones y siguientes pasos
10:25	Evaluación de la reunión	Todos	Puntos favorable y desfavorable
10:30	Cierre		

* **Líder** de la discusión

Participantes: G. Cuevas, M. García, A. Amescua, J. A. Cerrada, J. García, J.A. Calvo-Manzano, J.F. Estivariz

Facilitador - J.A. Cerrada
Notario - J.A. Calvo-Manzano
Controlador- M. García

En un proceso de toma de decisiones el equipo es más eficaz que el individuo. Para realizar un proceso de toma de decisiones eficaz es necesario:

- 1) Realizar un análisis previo objetivo y realista
- 2) Definir el problema implícito
- 3) Seguir un proceso disciplinado de razonamiento.

En este último paso los cuatro métodos más comunes son los siguientes:

- **Consenso:** encontrar una propuesta lo más aceptable posible de manera que todos los miembros puedan apoyarla y ningún miembro se oponga. cuando las decisiones son importantes, tienen grandes ramificaciones o afectan a mucha gente.
- **Votación:** cuando se ha determinado que la mayoría puede manejar la implementación sin la implicación activa de aquellos que pierden en el voto.
- **Subgrupo:** cuando un subgrupo tiene la información y conocimientos necesarios para tomar la decisión (consenso a menor escala).
- **Una persona:** ante una emergencia, cuando hay plena confianza o cuando el resultado sólo impacta al que toma la decisión.

Roles en equipo

Líder: organiza y motiva al equipo. Hace un seguimiento de todas las tareas y lleva el control del cuaderno del proyecto. Resuelve los problemas y prepara el orden de cada reunión.

Responsable de desarrollo: es el encargado de dirigir la parte del desarrollo de software. Guiará el equipo hacia las mejores decisiones a tomar respecto al software que desarrollar. Diseño de alto nivel, diseño de bajo nivel, implementación y pruebas.

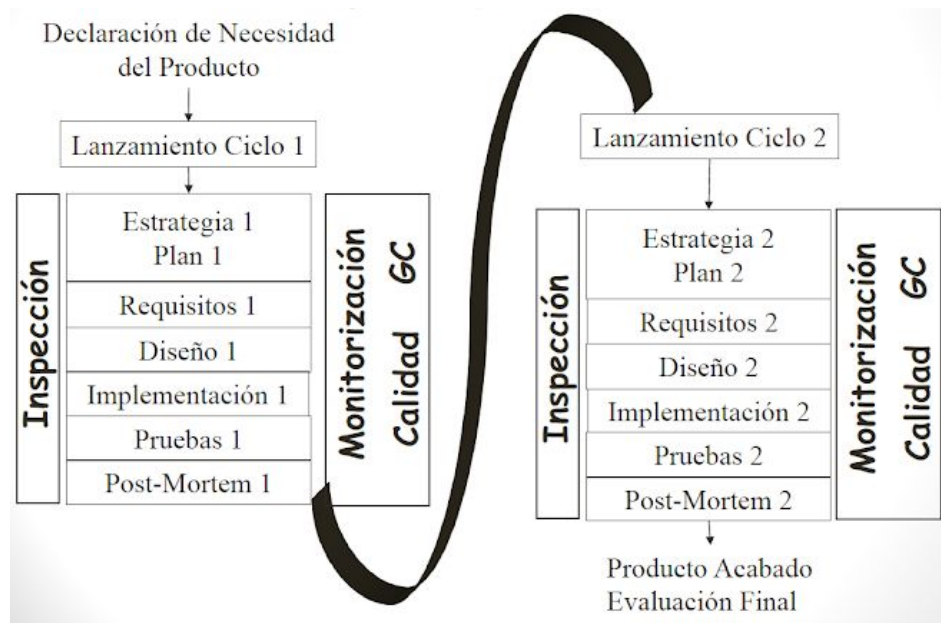
Responsable de planificación: facilita la planificación de las tareas de cada miembro del equipo. Produce así informes sobre las actividades desarrolladas por cada uno de ellos y asegura un correcto equilibrio de carga de trabajo.

Responsable de calidad/proceso: es el encargado que el trabajo de todos los ingenieros produzcan trabajo de calidad. Tiene la responsabilidad de asegurar que se implementen y mantengan los procesos para gestionar la calidad.

Responsable de soporte: es el encargado de facilitar el trabajo de los ingenieros. Aporta herramientas de trabajo al equipo y archiva la documentación.

Ciclo de vida (CV)

Es un marco de trabajo definido en ingeniería del software en equipo. Describe cómo planificar y gestionar un proyecto así como las responsabilidades de cada uno de los roles. Se da énfasis a los procesos, al producto y al trabajo en equipo. Establece también medidas estándar de calidad y rendimiento.



1. Lanzamiento

- 1.1. Asignar los equipos y roles
- 1.2. Elaboración de plantillas de documentos y estándar de documentación
- 1.3. Elaboración de estándar de Proceso de Desarrollo

2. Estrategia y Planificación

- 2.1. Elaboración de la lista de requisitos
- 2.2. Establecer la estrategia de desarrollo
- 2.3. Hacer estimaciones de tamaño y de esfuerzo
- 2.4. Evaluar los riesgos
- 2.5. Realizar los plane del equipo e individuales del ciclo 1

3. Requisitos

- 3.1. Revisión de la declaración de necesidades
- 3.2. Clarificación de la declaración de necesidades
- 3.3. Asignación de tareas a los requisitos
- 3.4. Documentación de los requisitos (documento ERS)
- 3.5. Plan de pruebas del sistema (PPS)
- 3.6. Inspección de los requisitos y del plan de pruebas del sistema
- 3.7. Línea básica de requisitos (ERS y PPS)

4. Diseño

- 4.1. Estándares Fase de diseño y Estándar Doc diseño (DAN Y DBN)
- 4.2. Asignación de tarea del diseño (DAN)
- 4.3. Especificación de diseño (documento DAN)
- 4.4. Plan de pruebas de integración (PPI)
- 4.5. Inspección del diseño y del plan de pruebas de integración
- 4.6. Línea básica del diseño (DAN y PPI)

5. Implementación

- 5.1. Asignación de tareas
- 5.2. Diseño de bajo nivel (Documento DBN)
- 5.3. Inspección de diseño de bajo nivel
- 5.4. Línea básica del DBN (DBN)
- 5.5. Codificación
- 5.6. Inspección de código
- 5.7. Línea básica del código (módulos/funciones de código)
- 5.8. Pruebas unitarias (casos y resultados de las PU)

6. Pruebas

- 6.1. Construcción
- 6.2. Pruebas de integración (casos y resultados de las PI)
- 6.3. Pruebas del sistema (casos y resultados de las PS)

7. Post-mortem

- 7.1. Visión general del proceso post-mortem
- 7.2. Revisión de datos del proceso
- 7.3. Preparar el informe del ciclo 1

Tema 2 - Estimación y planificación de proyectos de software

Gestión del proyecto

Utilización de las técnicas y actividades de gestión requeridas para conseguir un producto software de alta calidad, de acuerdo con las necesidades de los usuarios, dentro de un presupuesto y con una planificación de tiempos establecidos previamente.

- Estimación de proyectos: predicción de personal, esfuerzo, costes y planificación que se requerirá para realizar todas las actividades y construir todos los productos asociados con el proyecto.
- Planificación de proyectos: proceso de selección de una estrategia para la producción de unos productos finales dados, así como la definición de las actividades a realizar para conseguir ese objetivo, la concurrencia y solapamiento de dichas actividades. También se deben asignar recursos a las actividades anteriores en función del plan establecido.
- Seguimiento de proyectos: recolección de datos y su almacenamiento, sobre tiempos, recursos, costes, e hitos asociados con un proyecto, para el análisis y estudio de la evolución real de dicho proyecto, comparando el progreso real con el planificado.

En un diagrama de Gantt se expondrá el tiempo de dedicación previsto para las diferentes tareas o actividades a lo largo del tiempo atribuido al proyecto.

Gestión del proceso

Conjunto de técnicas y actividades que permiten una adecuada gestión de los procesos personales de los constructores y de los productos que participan en el proyecto.

Proyecto

Acción en la que recursos humanos, financieros y materiales se organizan de una nueva forma para acometer un trabajo único. En este trabajo, dadas unas especificaciones y dentro de unos límites de costes y tiempo, se intenta conseguir un cambio beneficioso dirigido por unos objetivos cualitativos y cuantitativos.

Estimación de software

Actividad que permite obtener principalmente respuestas aproximadas a las preguntas ¿cuánto costará? ¿cuánto tiempo llevará hacerlo?

Las respuestas a estas preguntas son complejas de obtener ya que entran a tener en cuenta muchos factores distintos, además que dos proyectos nunca serán iguales.

El estimador debe ser un profesional, que no tenga ningún interés, directo o indirecto, en los resultados del proceso de estimación y que está únicamente guiado por su profesionalidad.

Por lo general no es hasta bien adentro del proceso de desarrollo que se tiene la suficiente información para poder hacer una estimación correcta.

Métricas del software

Identificación y medidas de los parámetros esenciales que afectan al desarrollo del software. Se concentran en medidas del producto y medidas del proceso. Además serán objetivas o subjetivas según si el valor obtenido depende del juicio de los expertos y serán primitivas o calculadas según si se pueden obtener directamente.

Sirven para estimar tamaño y costes del software, controlar proyectos, predecir los niveles de calidad (errores, fiabilidad, mantenibilidad) y verificar el diseño de software.

Puntos de función (PF)

Técnica para medir el tamaño de un desarrollo en términos de la funcionalidad proporcionada al usuario final. Es una técnica independiente de la tecnología y sus resultados son consistentes.

Se realiza siguiendo los pasos a continuación:

1. Determinación del tipo de cuenta de PF
2. Identificación de los límites del sistemas
3. Cálculo de PF sin ajustar
4. Cálculo del factor de ajuste

5. Cálculo de PF ajustados

Para el cálculo de PF sin ajustar se tendrán en cuenta los siguientes tipos de función (cada uno de ellos vendrá clasificado por un distintivo BAJO - MEDIO - ALTO según su complejidad):

- Ficheros lógico internos (ILF): consecuencia de un requisito de usuario, son mantenidos por la aplicación. Por ejemplo ficheros maestros o tablas mantenidas por el usuario.
- Ficheros de interfaz externos (EIF): son como ILF pero son mantenidos por otra aplicación.
- Entradas externas (EI): datos o información de control que procede de fuera de los límites de la aplicación. Un ILF es mantenido mediante esa entrada externa. Por ejemplo transacciones, pantallas de entrada o información de control para creación de un informe.
- Salidas externas (EO): datos o información de control que sale de los límites de la aplicación. Por ejemplo informes, transferencia de datos a otra aplicación o gráficas. Las ayudas no son EO.
- Consultas externas (EQ): proceso elemental que obtiene una combinación de I/O como resultado de una recuperación de datos. No se mantiene ningún ILF. La petición llega desde fuera de los límites del sistema y la salida sobrepasa los límites del sistema. Por ejemplo consultas de usuario sin actualización, pantalla o mensaje de ayuda y menús de selección.

A continuación se valorará una serie de 14 características generales (puntuando cada una de ellas del 0 al 5 siendo el 0 el valor que menos influencia le da a la característica) y nuestro factor de ajuste será

$$AF = (TDI * 0.01) + 0.65$$

donde TDI es la suma de todas las puntuaciones de las características.

Finalmente

$$PF \text{ ajustados} = PF \text{ no ajustados} * AF$$

Las características generales son:

- 1) Comunicación de datos
- 2) Funciones distribuidas
- 3) Rendimiento
- 4) Configuraciones fuertemente utilizadas
- 5) Frecuencias de transacciones
- 6) Entrada de datos on-line
- 7) Eficiencia del usuario final

- 8) Actualización on-line de los ILF
- 9) Procesos complejos
- 10) Reusabilidad
- 11) Facilidad de instalación
- 12) Facilidad de operación
- 13) Instalación en distintos lugares
- 14) Facilidad de cambios

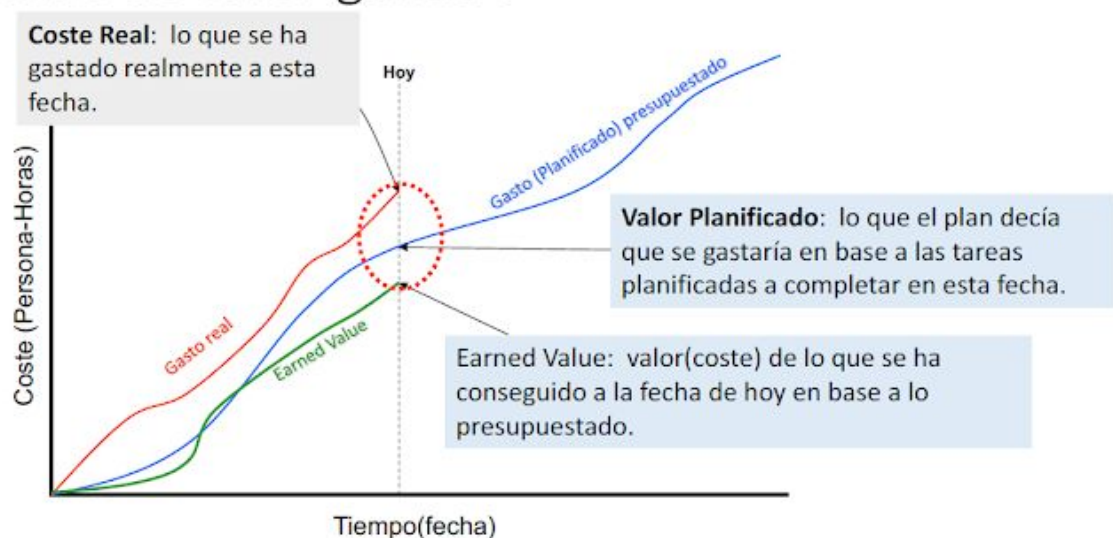
Valor ganado (EV - BCWP)

Esta medida permite a los desarrolladores y sus equipo medir el progreso del proyecto aún cuando completen las tareas en un orden diferente al planificado. Es la suma del valor planificado de todas las tareas de un proyecto.

El valor planificado (PV - BCWS) mide el porcentaje de cada tarea del proyecto sobre el total del proyecto. De esta manera, una tarea planificada de 1'5 horas en un proyecto de 1000 horas debería tener un PV de 0'15.

Así para valorar el estado de un proyecto será suficiente con medir el EV acumulado hasta la fecha y compararlo con el PV acumulado hasta esa fecha.

Gráfica de valor ganado



La principal ventaja sobre otros métodos de control de proyectos es el seguimiento continuo del valor ganado y otros indicadores permite al jefe de proyecto predecir cuando se logrará el éxito, y los riesgos de estar fuera de plazo, presupuesto y otros.

La varianza en el coste (CV) será la diferencia entre EV y AC (valor ganado y recursos gastados). El índice de rendimiento de coste (CPI) es EV / AC .

La varianza en el calendario (SV) será la diferencia entre EV y PV (valor ganado y valor planificado). El índice de rendimiento de calendario (SPI) es EV / PV .

El presupuesto final (BAC) es la suma de todos los PV.

La estimación del presupuesto final (EAC) es igual a la división entre BAC y CPI.

La varianza a la finalización (VAC) es la diferencia entre el BAC y el EAC.

Earned Value: Hoja datos

Schedule Variance	$SV = EV - PV$	Negativo significa retraso
Schedule Performance Index	$SPI = EV / PV$	< 1 significa retraso
Cost Variance	$CV = EV - AC$	Negativo significa sobre costes.
Cost Performance Index	$CPI = EV / AC$	< 1 significa sobre coste

Estimate At Completion	$EAC = BAC / CPI$	Previsión de coste final
Estimate At Completion	$EAC = AC + ETC$	Coste actual + previsto
Variance At Completion	$VAC = BAC - EAC$	Qué margen tenemos?
To Complete Performance Index	$TCPI_{BAC} = (BAC - EV) / (BAC - AC)$	Podemos alcanzarlo?

Cocomo

Software que permite estimación de calendario y esfuerzo en un proyecto de desarrollo de software. Puntos de función, factores de escala y multiplicadores de esfuerzo.

FACTORES DE ESCALA

Factores de escala	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto
PREC Predente del equipo	Enormemente sin precedencia	Largamente sin precedencia	En cierto modo sin precedencia	Bastante familiar	Largamente familiar	Enormemente familiar
FLEX Flexibilidad desarr.	riguroso	Ocasionalmente relajado	Alguna relajación	Conformidad general	Alguna conformidad	Metas generales
RESL Resolu. Riesgos	Pequeño 20%	Poco 40%	A menudo 60%	Generalmente 75%	Principalmente 90%	Completo 100%
TEAM Cohesion del equipo	Interacciones muy dificultosas	Pocas interacciones dificultosas	Básicamente interacciones cooperativas	Mayoritaria-mente cooperativas	Altamente cooperativa	Interacciones completas
PMAT Madurez	*	*	*	*	*	*

MULTIPLICADORES DEL ESFUERZO POST ARQUITECTURA

PRODUCTO

RELY= fiabilidad requerida por el SW

DATA= tamaño de la base de datos

DOCU=documentación requerida

CPLX= complejidad del producto

RUSE= reusabilidad requerida

PLATAFORMA

TIME= restricciones de tiempo de ejecución

STOR= restricciones de almacenamiento

PVOL= volatilidad de la plataforma

PERSONAL

ACAP= capacidad de análisis

AEXP= experiencia

PCAP= capacidades de programación

PEXP= experiencia en la plataforma

LEXP= experiencia en el lenguaje de programación

PCON= continuidad del personal

PROYECTO

TOOL= uso de herramientas de SW

SITE= desarrollo en diferentes sitios

Tema 3 - Gestión de configuración

Los cambios descontrolados pueden hacer desperdiciar gran cantidad de tiempo, degradar los diseños, originar reparaciones mal aplicadas, y que incluso dan por resultado productos que se van a desechar.

Un ingeniero podía actualizar un módulo con una reparación de los defectos mientras que otro modificaba una característica del mismo módulo al mismo tiempo. Cuando cada ingeniero devuelva su versión a la librería del sistema, es probable que un ingeniero sobrescriba y destruya el trabajo del otro ingeniero.

Plan de GCS

El responsable de soporte realiza el plan de gestión de configuración, define el formulario de petición de cambio de configuración y define el formulario de informe de estado de la configuración.

Define también el Comité de Control de Configuración (CCC) y sus procedimientos, especifica las herramientas de soporte y facilidades necesarias, revisa con el equipo los procedimientos para su conformidad.

El funcionamiento general del GCS es mantener copias de todas las versiones de cada producto y registrar todos los cambios hechos a la línea base del producto. De esta forma se conoce los contenidos del producto y existen herramientas para dar marcha atrás.

La línea base es el conjunto de documentos y otros materiales que representan de manera oficial el producto en cualquier momento. Una vez que un elemento del producto ha sido puesto en la línea base nadie puede cambiarlo sin la autorización del CCC y el formulario de petición de cambios correspondiente.

Los productos se ponen generalmente bajo control de configuración después de que hayan sido inspeccionados, y corregidos todos los problemas que se identificaron.

Pasos para un EC nuevo

- 1) Se propone al CCC añadir un producto a la línea base a través de una petición de cambios de configuración (CCR).
- 2) El CCC revisa la entrega, abre petición de cambio (PC), le asigna un identificador y pasa la petición al departamento de Calidad.
- 3) El responsable de calidad/proceso confirma la calidad el producto.
- 4) El CCC revisa la entrega y la aprueba (reunión final), solicita más información o la desaprueba.

Pasos para un EC existente

- 1) Se propone al CCC modificar un producto de la línea base a través de una petición de cambios de configuración (CCR).
- 2) El CCC revisa la entrega y la aprueba (reunión inicial), solicita más información o la desaprueba.
- 3) En caso de aprobación se solicita al autor de dicho producto que implemente el cambio.
- 4) Responsable de calidad/proceso confirma la calidad del cambio.
- 5) El CCC procede a realizar el cambio en la línea base (reunión final).

Formulario de solicitud de petición de cambio (PC)

Este formulario se emplea cuando se envía un elemento de configuración (EC) al comité de control de cambios (CCC) para su inclusión en la línea base del producto.

Formulario de informe de estado de configuración (IEC)

Este formulario se utiliza para proporcionar semanalmente el estado de la información el sistema de gestión de configuración software.

Proceso de inspección

- 1) Revisión personal realizada por el productor (desarrollador) del producto (cuando da el visto bueno se rellena la fecha de aprobación en el formulario de PC).
- 2) El Jefe de Calidad/Proceso es el moderador de la inspección, aunque lo acompañan 1 o 2 personas más.
- 3) Inicialmente cada uno realiza una revisión individual para encontrar cualquier problema o área de confusión. Cada revisor anota el tiempo gastado en la revisión y marca el defecto en el material. No se registra el LOGD aún debido a que puede que un mismo error sea detectado por varios revisores.
- 4) Se realiza la reunión. Se va línea a línea. El moderador registra los defectos graves en el formulario INS y el productor en el formulario LOGD. Los defectos leves sólo se registran en el formulario INS.
- 5) Al terminar la reunión se estima el número de defectos que es probable que aparezcan después de la reunión y se calcula el llamado "rendimiento estimado de la inspección".

$$\begin{aligned} \text{Rendimiento Estimado de la Inspección} &= \frac{100 * (\text{Total defectos encontrados})}{(\text{Total defectos estimados})} \\ &= \frac{100 * (A + B - C)}{(A * B / C)} \end{aligned}$$

(siempre se redondea cifras hacia arriba)

- 6) Finalmente si el rendimiento de la inspección es menor que 85% es necesario volver a hacer una inspección o replantearse los métodos de su realización para el próximo ciclo.

Formulario INS

Nombre	Defectos		Datos Preparación			Rendimiento Estimado
	Grave	Leve	Tamaño	Tiempo (minutos)	Tasa	
Ingeniero 1	7	3	380	85	268	58,3
Ingeniero 2	9	2	380	97	235	75,0
Ingeniero 3	5	4	380	68	335	41,7
Totales	21	9	1140	250	274	91,7

El tamaño es típicamente Lines of Code (LOC), la tasa representa el ratio de LOC / horas (Tamaño/tiempo * 60). El rendimiento estimado se calcula con la fórmula anterior involucrando A, B y C.

No	Descrip Defecto	Defectos		Ingenieros que encuentran los defectos graves					
		Grave	Leve	Ing 1	Ing 2	Ing 3		A	B
1	Convertidor	1			1	1		1	1
9	>=	1		1	1			1	1
12		1		1					1
13		1		1	1			1	1
15		1			1			1	
19		1		1	1	1		1	1
29		1		1		1			1
31		1		1	1	1		1	1
35		1			1			1	
37		1			1	1		1	1
41		1		1	1			1	1
Totales		11		7	9	5		9	9
Defectos únicos				1	2	0		2	2

A = se marca uno cuando el ingeniero que más defectos haya encontrado ha encontrado ese defecto en cuestión.

B = se marca cuando al menos uno de los otros dos ingenieros ha encontrado ese defecto en cuestión.

C = número de defectos comunes encontrados

(sólo se cuentan defectos graves)

Tema 4 - Gestión de calidad

Economía de la calidad

El software controla la mayoría de los negocios, gobiernos y sistemas militares.

El coste, los plazos y la calidad del software es ahora mismo una preocupación crítica de los negocios.

El problema es que los proyectos de software no son escalables. Los proyectos grandes fallan porque los compromisos no se ajustan a la realidad, son más difíciles de controlar, hay mayor número de problemas de calidad y los equipos necesitan un liderazgo y entrenamiento impecable.

La calidad es básicamente ajustarse a las necesidades de los usuarios. Estas necesidades comprenden:

- Realizar las tareas requeridas
- Cumplir con requisitos de rendimiento
- Poderse utilizar y ser conveniente
- Ser económico y estar a tiempo
- Ser fiable y confiable

La calidad de un sistema software está condicionada por la calidad del peor de sus componentes. La calidad de un componente software está condicionada por el individuo que lo desarrolló. La calidad de un componente software está condicionada por el proceso del proceso que se usó para desarrollarla.

Se estima que +50% del tiempo de un programador es empleado en realizar pruebas para asegurar el correcto funcionamiento del programa. Se plantea gestionar la calidad y así conseguir que únicamente el 20% del tiempo sea dedicado a pruebas, lo que provoca una mayor productividad.

Las principales maneras de encontrar y corregir defectos son:

- Compilar
- Pruebas unitarias
- Pruebas de integración y sistemas
- Inspecciones
- Revisiones personales

Control de calidad de software

Se plantea que las revisiones son más eficientes que las pruebas (cosa que a nivel personal dudo). Las revisiones siguen un proceso disciplinado con guías, listas de comprobación y estándares que garantizan encontrar los defectos. Es común seguir una checklist para comprobar que todo se cumple.

Además de hacer revisiones sobre el código es muy recomendable hacer revisiones sobre el diseño del programa igualmente. Un buen diseño facilita la revisión de código a posteriori.

Gestión de calidad de software

Se emplean distintos tipos de métricas de calidad como el rendimiento, velocidad de revisión, defectos encontrados por unidad de tamaño, defectos inyectados y eliminados por hora, nivel de eliminación de defectos o coste de calidad.

Lo más común es contrastar el rendimiento (porcentaje de defectos encontrados en una revisión frente al total de defectos existentes) con la velocidad (en general LOC/hora).

El principal foco de las inspecciones debería ser encontrar problemas que se han omitido

Cuando los programas tienen muchos defectos simples, los inspectores se distraen y normalmente omiten problemas importantes.

Con un plan de calidad se pretende responder a preguntas como:

- ¿Cómo estamos fabricando?
- ¿Cómo de bien funcionan las inspecciones y revisiones?
- ¿Qué densidad tengo en pruebas?
- ¿Qué nivel de esfuerzo dedico?
- ¿Me paso en revisiones?
- ¿A qué marcha vamos revisando?
- ¿A qué marcha se introducen defectos?
- ¿A qué marcha se eliminan defectos?
- ¿Dónde se eliminan más defectos? ¿Inspecciones? ¿Revisiones? ¿Pruebas?
- ¿En qué fase fue inyectado un defecto?

En todo caso normalmente es caro encontrar defectos, y encima luego se tienen que corregir. La prevención de defectos representa pues un gran ahorro y es algo muy interesante conseguir.

En primer paso en un proceso de prevención es ajustar la prioridad a cierto tipo de defecto que son más frecuentes, más problemáticos o más fáciles de prevenir. Luego se entra en un ciclo continuo de ajuste y mejora de las acciones de prevención hasta con dar con un método que funcione. Algunas de las técnicas utilizadas son los estándares de tipos de defecto para la clasificación de los defectos encontrados y el diagrama de Pareto para detectar qué defectos son responsables del mayor número de problemas.

Ejercicios PC tipo examen

1. Teniendo en cuenta los siguientes datos de una Petición de Cambio sobre un Elemento de Configuración nuevo (no existente en Línea Base), buscar los defectos / inconsistencias que se encuentren en ella, considerando que los datos del Estado de la PC son correctos.

Estado de la PC			
Estado (márquese según vaya procediendo)	Fecha		
Abierta: <input checked="" type="checkbox"/>	17 Mayo 2016		
Aprobada: <input checked="" type="checkbox"/>	22 Mayo 2016		
Desaprobada: <input type="checkbox"/>			
En Implementación: <input type="checkbox"/>			
Cerrada: <input checked="" type="checkbox"/>	29 Mayo 2016		
Aprobaciones			
Propietario del EC	JAC	Fecha:	18 Mayo 2016
Responsable de Calidad/Proceso	TSF	Fecha:	23 Mayo 2016
CCC (reunión inicial)	GCA	Fecha:	19 Mayo 2016
CCC (reunión final)	GCA	Fecha:	29 Mayo 2016

La fecha del propietario del EC tiene que ser igual o anterior a la fecha de apertura de la PC.

La fecha de la aprobación de Calidad tiene que ser anterior a la fecha de aprobación del CCC.

La fecha de la reunión inicial tiene que ser la misma que la fecha de apertura de la PC.

La fecha de la reunión final tiene que ser la misma que la fecha de aprobación de la PC.

2. Teniendo en cuenta los siguientes datos de una Petición de Cambio sobre un Elemento de Configuración ya existente en Línea Base, buscar los defectos / inconsistencias que se encuentren en la parte de "Aprobaciones", considerando que los datos correspondientes al "Estado de la PC" son correctos y que el CCC tiene que aprobar la PC tanto al principio cuando recibe la PC (indicado con el campo de estado de "aprobada") como una vez que se haya implementado debido a la criticidad de la PC (indicado con el campo de estado de "cerrada").

Estado de la PC			
Estado (márquese según vaya procediendo)	Fecha		
Abierta: <input checked="" type="checkbox"/>	10 Marzo 2007		
Aprobada: <input checked="" type="checkbox"/>	15 Marzo 2007		
Desaprobada: <input type="checkbox"/>			
En Implementación: <input checked="" type="checkbox"/>	18 Marzo 2007		
Cerrada: <input checked="" type="checkbox"/>	22 Marzo 2007		
Aprobaciones			
Propietario del EC	JAC	Fecha:	17 Marzo 2007
Responsable de Calidad/Proceso	TSF	Fecha:	20 Marzo 2007
CCC (Reunión inicial)	GCA	Fecha:	18 Marzo 2007
CCC (Reunión final)	GCA	Fecha:	22 Marzo 2007

La fecha del propietario del EC tiene que ser igual o después a la fecha de implementación.

La fecha de la reunión inicial tiene que coincidir que la fecha de apertura???? y aprobación de la PC.

3. Indique el orden de las aprobaciones que sigue una Petición de Cambio sobre un elemento de configuración ya existente en línea base, con respecto a las partes interesadas indicadas en la siguiente tabla (póngase al lado del campo fecha el número de orden, donde un 1 indica que es el primero en aprobar, un 2 el segundo, y así sucesivamente).

Aprobaciones			
Propietario del EC	JACMV	Fecha:	
Responsable de Calidad/Proceso	TSFG	Fecha:	
CCC(reunión inicial)	GCA	Fecha:	
CCC (reunión final)	GCA	Fecha:	

2
3
1
4

4. Teniendo en cuenta los siguientes datos de una Petición de Cambio sobre un Elemento de Configuración existente en Línea Base, buscar los defectos / inconsistencias que se encuentren en ella, considerando que los datos del Estado de la PC son correctos.

Estado de la PC	
Estado (márquese según vaya procediendo)	Fecha
Abierta: <input checked="" type="checkbox"/>	17 Junio 2008
Aprobada: <input checked="" type="checkbox"/>	22 Junio 2008
Desaprobada: <input type="checkbox"/>	
En Implementación: <input checked="" type="checkbox"/>	24 Junio 2008
Cerrada: <input checked="" type="checkbox"/>	29 Junio 2008

Aprobaciones			
Propietario del EC	JACMV	Fecha:	17 Junio 2008
Responsable de Calidad/Proceso	TSFG	Fecha:	23 Junio 2008
CCC(reunión inicial)	GCA	Fecha:	22 Junio 2008
CCC (reunión final)	GCA	Fecha:	29 Junio 2008

La fecha de reunión inicial tiene que coincidir con la fecha de apertura y aprobación de la PC.

La fecha de aprobación del propietario debe ser igual o posterior a la fecha de implementación.

La fecha de calidad tiene que ser posterior a la fecha de la aprobación del propietario.

5. Teniendo en cuenta los siguientes datos de una Petición de Cambio sobre un Elemento de Configuración NUEVO en Línea Base, buscar los defectos / inconsistencias que se encuentren en ella, considerando que los datos del Estado de la PC son correctos.

Estado de la PC			
Estado (márquese según vaya procediendo)	Fecha		
Abierta: <input checked="" type="checkbox"/>	4 Junio 2009		
Aprobada: <input checked="" type="checkbox"/>	6 Junio 2009		
Desaprobada: <input type="checkbox"/>			
En Implementación: <input type="checkbox"/>			
Cerrada: <input checked="" type="checkbox"/>	10 Junio 2009		
Aprobaciones			
Propietario del EC	JAC	Fecha:	4 Junio 2009
Responsable de Calidad/Proceso	TSF	Fecha:	7 Junio 2009
CCC(reunión inicial)	GCA	Fecha:	4 Junio 2009
CCC (reunión final)	GCA	Fecha:	10 Junio 2009

La fecha de reunión inicial tiene que coincidir con la fecha de aprobación de la PC.

La fecha de calidad tiene que ser anterior a la fecha de aprobación de la PC.

6. Teniendo en cuenta los siguientes datos de una Petición de Cambio sobre un Elemento de Configuración EXISTENTE en Línea Base, buscar los defectos / inconsistencias que se encuentren en ella, considerando que los datos del ESTADO DE LA PC son CORRECTOS. (1 punto)

Estado de la PC			
Estado (márquese según vaya procediendo)	Fecha		
Abierta: <input checked="" type="checkbox"/>	3 Enero 2016		
Aprobada: <input checked="" type="checkbox"/>	6 Enero 2016		
Desaprobada: <input type="checkbox"/>			
En Implementación: <input checked="" type="checkbox"/>	7 Enero 2016		
Cerrada: <input checked="" type="checkbox"/>	9 Enero 2016		
Aprobaciones			
Propietario del EC	JAC	Fecha:	6 Enero 2016
Responsable de Calidad/Proceso	TSF	Fecha:	8 Enero 2016
CCC(reunión inicial)	AM	Fecha:	8 Enero 2016
CCC (reunión final)	AM	Fecha:	9 Enero 2016

La fecha de reunión inicial tiene que coincidir con la fecha de aprobación.

La fecha de la aprobación del propietario tiene que ser igual o posterior a la fecha de implementación.

7. Teniendo en cuenta los siguientes datos de una Petición de Cambio sobre un Elemento de Configuración EXISTENTE en Línea Base, buscar los defectos / inconsistencias que se encuentren en ella, considerando que los datos del ESTADO DE LA PC son CORRECTOS (1 punto).

Estado de la PC			
Estado (márquese según vaya procediendo)	Fecha		
Abierta: <input checked="" type="checkbox"/>	7 Julio 2016		
Aprobada: <input checked="" type="checkbox"/>	7 Julio 2016		
Desaprobada: <input type="checkbox"/>			
En Implementación: <input checked="" type="checkbox"/>	8 Julio 2016		
Cerrada: <input checked="" type="checkbox"/>	10 Julio 2016		
Aprobaciones			
Propietario del EC	JACM	Fecha:	6 Julio 2016
Responsable de Calidad/Proceso	TSF	Fecha:	9 Julio 2016
CCC(reunión inicial)	AM	Fecha:	10 Julio 2016
CCC (reunión final)	AM	Fecha:	10 Julio 2016

La fecha de reunión inicial tiene que coincidir con la fecha de aprobación de la PC.
La fecha de la aprobación del propietario tiene que ser posterior a la fecha de implementación.

8. Teniendo en cuenta los siguientes datos de una Petición de Cambio sobre un Elemento de Configuración EXISTENTE en Línea Base, buscar los defectos / inconsistencias que se encuentren en ella, considerando que los datos del ESTADO DE LA PC son CORRECTOS. (1 punto)

Estado de la PC			
Estado (márquese según vaya procediendo)	Fecha		
Abierta: <input checked="" type="checkbox"/>	6 Mayo 2016		
Aprobada: <input checked="" type="checkbox"/>	6 Mayo 2016		
Desaprobada: <input type="checkbox"/>			
En Implementación: <input checked="" type="checkbox"/>	7 Mayo 2016		
Cerrada: <input checked="" type="checkbox"/>	9 Mayo 2016		
Aprobaciones			
Propietario del EC	JACM	Fecha:	5 Mayo 2016
Responsable de Calidad/Proceso	TSF	Fecha:	8 Mayo 2016
CCC (reunión inicial)	AM	Fecha:	9 Mayo 2016
CCC (reunión final)	AM	Fecha:	9 Mayo 2016

La fecha de reunión inicial tiene que coincidir con la fecha de aprobación de la PC.
La fecha de aprobación del propietario tiene que ser igual o posterior a la fecha de implementación.